Package 'RNifti'

October 19, 2018

Version 0.10.0

Date 2018-10-19

Title Fast R and C++ Access to NIfTI Images

Imports Rcpp (>= 0.11.0)

Suggests testthat (>= 0.11.0), covr, reportr

Enhances oro.nifti, tractor.base

LinkingTo Rcpp

Description Provides very fast read and write access to images stored in the NIfTI-1 and ANALYZE-7.5 formats, with seamless synchronisation between compiled C and interpreted R code. Also provides a C/C++ API that can be used by other packages. Not to be confused with 'RNiftyReg', which performs image registration.

License GPL-2

URL https://github.com/jonclayden/RNifti

BugReports https://github.com/jonclayden/RNifti/issues

Encoding UTF-8

RoxygenNote 6.0.1

NeedsCompilation yes

Author Jon Clayden [cre, aut],
Bob Cox [aut],
Mark Jenkinson [aut],
Matt Hall [ctb],
Rick Reynolds [ctb],
Kate Fissell [ctb],
Jean-loup Gailly [cph],
Mark Adler [cph]

Maintainer Jon Clayden <code@clayden.org>

Repository CRAN

Date/Publication 2018-10-19 14:50:02 UTC

R topics documented:

`dim.internalImage` *Internal images*

Description

An internal image is a simple R object with a few attributes including a pointer to an internal C structure, which contains the full image data. They are used in the package for efficiency, but can be converted to a normal R array using the `as.array` method. Attributes of these objects should not be changed.

Usage

```
## S3 method for class 'internalImage'
dim(x)

## S3 replacement method for class 'internalImage'
dim(x) <- value

## S3 method for class 'internalImage'
as.array(x, ...)
```

Arguments

`x`	An "`internalImage`" object.
`value`	Not used. Changing the dimensions of an internal image is invalid, and will produce an error.
`...`	Additional parameters to methods. Currently unused.

Author(s)

Jon Clayden <code@clayden.org>

ndim *Number of dimensions*

Description

This function is shorthand for `length(dim(object))`.

Usage

```
ndim(object)
```

Arguments

`object`	An R object.

Value

The dimensionality of the object. Objects without a `dim` attribute will produce zero.

Author(s)

Jon Clayden <code@clayden.org>

Examples

```
ndim(array(0L, dim=c(10,10)))
```

niftiHeader *Dump or construct a raw NIfTI or ANALYZE header*

Description

These functions extract the contents of a NIfTI-1 or ANALYZE-7.5 header, closely approximating how it is (or would be) stored on disk. Defaults will be used where information is missing, but no processing is performed on the metadata.

Usage

```
niftiHeader(image = list())

analyzeHeader(image = list())

## S3 method for class 'niftiHeader'
print(x, ...)

## S3 method for class 'analyzeHeader'
print(x, ...)
```

Arguments

image	An image, in any acceptable form (see retrieveNifti). A list containing partial header information is acceptable, including an empty list, which returns defaults for every field.
x	A "niftiHeader" object.
...	Ignored.

Details

The NIfTI-1 standard was originally formulated as a roughly backwards- compatible improvement on the ANALYZE format. Both formats use a binary header structure of 348 bytes, but the field names and their interpretation is often non-equivalent. These functions dump these fields, without regard to whether or not the result makes proper sense.

dumpNifti is an alias of niftiHeader, but the former is now soft-deprecated.

Value

For niftiHeader, a list of class "niftiHeader", with named components corresponding to the elements in a raw NIfTI-1 header. For analyzeHeader, the equivalent for ANALYZE-7.5.

Note

Several medical image analysis packages, such as SPM and FSL, use the ANALYZE originator field to store a coordinate origin. This interpretation is also returned, in the origin field.

Author(s)

Jon Clayden <code@clayden.org>

References

The NIfTI-1 standard (http://www.nitrc.org/docman/view.php/26/64/nifti1.h).

See Also

niftiVersion

Examples

```
niftiHeader(system.file("extdata", "example.nii.gz", package="RNifti"))

# Default header for a standard R array
niftiHeader(array(0L, dim=c(10,10)))
```

`niftiVersion` *Check the format version of a file*

Description

This function identifies the likely NIfTI format variant used by one or more files on disk.

Usage

```
niftiVersion(file)
```

Arguments

`file`	A character vector of file names.

Value

A vector of integers, of the same length as `file`. Each element will be 0 for ANALYZE format (the precursor to NIfTI-1), 1 for NIfTI-1 (which is now most common), 2 for NIfTI-2, or -1 if the file doesn't exist or doesn't look plausible in any of these formats.

Note

NIfTI-2 format, mostly a variant of NIfTI-1 with wider datatypes used for many fields, is not currently supported for reading, but it is detected by this function.

Author(s)

Jon Clayden <code@clayden.org>

See Also

`readNifti`, `niftiHeader`

Examples

```
path <- system.file("extdata", "example.nii.gz", package="RNifti")
niftiVersion(path)       # 1
```

pixdim *Pixel dimensions and units*

Description

By default, these generic functions return or replace the "pixdim" and "pixunits" attributes of their arguments. These represent the physical step size between pixel or voxel centre points, and the spatial and temporal units that they are given in. The former defaults to 1 in each dimension, if there is no attribute.

Usage

```
pixdim(object)

## Default S3 method:
pixdim(object)

pixdim(object) <- value

## Default S3 replacement method:
pixdim(object) <- value

pixunits(object)

## Default S3 method:
pixunits(object)

pixunits(object) <- value

## Default S3 replacement method:
pixunits(object) <- value
```

Arguments

object	An R object, generally an image.
value	Numeric vector of pixel dimensions along each axis, or character vector of abbreviated units. For dimensions, a scalar value will be recycled if necessary.

Value

pixdim returns a numeric vector of pixel dimensions. pixunits returns a character vector of length up to two, giving the spatial and temporal unit names.

Author(s)

Jon Clayden <code@clayden.org>

Examples

```
im <- readNifti(system.file("extdata", "example.nii.gz", package="RNifti"))
pixdim(im)
pixunits(im)
```

`readNifti` *Read a NIfTI-1 format file*

Description

This function reads one or more NIfTI-1 or ANALYZE-7.5 files into R, using the standard NIfTI-1 C library.

Usage

```
readNifti(file, internal = FALSE, volumes = NULL)
```

Arguments

`file`	A character vector of file names.
`internal`	Logical value. If `FALSE` (the default), an array of class `"niftiImage"`, containing the image pixel or voxel values, will be returned. If `TRUE`, the return value will be an object of class `"internalImage"`, which contains only minimal metadata about the image. Either way, the return value has an attribute which points to a C data structure containing the full image.
`volumes`	An integer vector giving the volumes to read (counting along all dimensions beyond the third jointly), or `NULL`, the default, in which case every volume is read. This cannot currently be set differently for each file read.

Value

An array or internal image, with class `"niftiImage"` (and possibly also `"internalImage"`), or a list of such objects if `file` has length greater than one.

Note

If the `internal` argument is `FALSE` (the default), the data type of the image pointer will be set to match one of R's native numeric data types, i.e., 32-bit signed integer or 64-bit double-precision floating-point. In these circumstances the data type reported by the `niftiHeader` function will therefore not, in general, match the storage type used in the file. See also the `datatype` argument to `writeNifti`.

Author(s)

Jon Clayden <code@clayden.org>

References

The NIfTI-1 standard (http://www.nitrc.org/docman/view.php/26/64/nifti1.h).

See Also

writeNifti

Examples

```
path <- system.file("extdata", "example.nii.gz", package="RNifti")
readNifti(path)
readNifti(path, internal=TRUE)
```

---

retrieveNifti *Obtain an internal NIfTI representation of an object*

Description

This function converts filenames, arrays and other image classes into an object of class "internalImage".

Usage

```
retrieveNifti(object)
```

Arguments

object	Any suitable object (see Details).

Details

If the object has an internal NIfTI pointer, that will be retrieved directly. Otherwise, if it is a string, it will be taken to be a filename. If it looks like a "nifti" object (from package oro.nifti), or an "MriImage" object (from package tractor.base), a conversion will be attempted. A list will be assumed to be of the form produced by niftiHeader. Finally, a numeric array or matrix will be converted using default image parameters.

Value

An internal image.

Author(s)

Jon Clayden <code@clayden.org>

See Also

readNifti, updateNifti

updateNifti *Update an internal NIfTI-1 object using a template*

Description

This function adds or updates the internal NIfTI-1 object for an array, using metadata from the template. The dimensions and, if available, pixel dimensions, from the `image` will replace those from the template.

Usage

```
updateNifti(image, template = NULL, datatype = "auto")
```

Arguments

`image`	A numeric array.
`template`	An image, in any acceptable form (see `retrieveNifti`), or a named list of NIfTI-1 properties like that produced by `niftiHeader`. The default of `NULL` will have no effect.
`datatype`	The NIfTI datatype to use within the internal image. The default, `"auto"` uses the R type. Other possibilities are `"float"`, `"int16"`, etc., which may be preferred to reduce object size. However, no checks are done to ensure that the coercion maintains precision, and this option is for advanced usage only.

Details

If `template` is a complete list of NIfTI-1 header fields, like that produced by `niftiHeader`, or an image, then it will be used to create the internal object, and then the data and metadata associated with the `image` will overwrite the appropriate parts. If `template` is an incomplete list, the `image` will be used to create the internal object, and then the specified fields will be overwritten from the list. This allows users to selectively update certain fields while leaving others alone (but see the note below).

Datatype information in a list `template` is ignored. The datatype can only be changed using the `datatype` argument, but in this case the internal object gets out of sync with the R array, so an internal image is returned to avoid the mismatch. Changing the internal datatype in this way is for advanced usage only.

Value

A copy of the original `image`, with its internal image attribute set or updated appropriately. If `datatype` is not `"auto"` then the result is an internal image.

Note

The `scl_slope` and `scl_inter` fields affect the numerical interpretation of the pixel data, so it is impossible in general to change them without also changing the array values on both the C and the R side. Therefore, to avoid unexpected side-effects, these fields are not affected by this function.

Author(s)

Jon Clayden <code@clayden.org>

`voxelToWorld` *Transform points between voxel and "world" coordinates*

Description

These functions are used to transform points from dimensionless pixel or voxel coordinates to "real-world" coordinates, typically in millimetres, and back. Actual pixel units can be obtained using the `pixunits` function. The `origin` function gives the voxel coordinates of the real-world origin.

Usage

```
voxelToWorld(points, image, simple = FALSE, ...)

worldToVoxel(points, image, simple = FALSE, ...)

origin(image, ...)
```

Arguments

`points`	A vector giving the coordinates of a point, or a matrix with one point per row.
`image`	The image in whose space the points are given, or a 4x4 numeric xform matrix.
`simple`	A logical value: if `TRUE` then the transformation is performed simply by rescaling the points according to the voxel dimensions recorded in the `image`. Otherwise the full xform matrix is used.
`...`	Additional arguments to `xform`.

Value

A vector or matrix of transformed points.

Note

Voxel coordinates are assumed by these functions to use R's indexing convention, beginning from 1.

Author(s)

Jon Clayden <code@clayden.org>

See Also

`xform`, `pixdim`, `pixunits`

Examples

```
im <- readNifti(system.file("extdata", "example.nii.gz", package="RNifti"))

# Find the origin
origin(im)
```

`writeNifti` *Write a NIfTI-1 format file*

Description

This function writes an image to NIfTI-1 format, using the standard NIfTI-1 C library.

Usage

```
writeNifti(image, file, template = NULL, datatype = "auto")
```

Arguments

`image`	An image, in any acceptable form (see `retrieveNifti`).
`file`	A character string containing a file name.
`template`	An optional template object to derive NIfTI-1 properties from. Passed to `updateNifti` if `image` is an array.
`datatype`	The NIfTI datatype to use when writing the data out. The default, `"auto"` uses the R type or, for internal images, the original datatype. Other possibilities are `"float"`, `"int16"`, etc., which may be preferred to reduce file size. However, no checks are done to ensure that the coercion maintains precision.

Author(s)

Jon Clayden <code@clayden.org>

References

The NIfTI-1 standard (http://www.nitrc.org/docman/view.php/26/64/nifti1.h).

See Also

`readNifti`, `updateNifti`

Examples

```
## Not run: writeNifti(im, "image.nii.gz", datatype="float")
```

`xform` *Obtain or replace the "xform" transforms for an image*

Description

These functions convert the "qform" or "sform" information in a NIfTI header to or from a corresponding affine matrix. These two "xform" mechanisms are defined by the NIfTI standard, and may both be in use in a particular image header. They define the relationship between the storage order of the image and real space.

Usage

```
xform(image, useQuaternionFirst = TRUE)

qform(x) <- value

sform(x) <- value

orientation(x, useQuaternionFirst = TRUE)

orientation(x) <- value

rotation(x, useQuaternionFirst = TRUE)
```

Arguments

`image, x`	An image, in any acceptable form (see `retrieveNifti`), or a 4x4 numeric xform matrix.
`useQuaternionFirst`	A single logical value. If `TRUE`, the "qform" matrix will be used first, if it is defined; otherwise the "sform" matrix will take priority.
`value`	A new 4x4 qform or sform matrix, or orientation string. If a matrix has a `"code"` attribute, the appropriate qform or sform code is also set.

Details

Image orientation is indicated using a three-character string, with each character indicating the approximate world-space direction of the positive axes in the first, second and third dimensions, in order. Each character may be 'R' for left-to-right, 'L' for right-to-left, 'A' for posterior-to- anterior, 'P' for anterior-to-posterior, 'S' for inferior-to-superior, or 'I' for superior-to-inferior. The default for NIfTI is RAS, meaning that the first dimension points towards the right, the second towards the front and the third towards the top. An xform matrix is an affine transform relative to that default.

The upper-left 3x3 matrix in a 3D affine transform governs scale, rotation and skew, while the last column is a translation. (The `rotation` function extracts the rotation part alone.) The final row is always (0,0,0,1). Reorienting an image involves permuting and possibly reversing some of the axes, both in the data and the metadata. The sense of the translation may also need to be reversed, but

this is only possible if the image dimensions are known, which isn't the case when reorienting an xform alone.

Value

For `xform`, an affine matrix corresponding to the "qform" or "sform" information in the image header. For `orientation`, a string with three characters indicating the (approximate) orientation of the image. The replacement forms return the modified object.

Note

The qform and sform replacement functions are for advanced users only. Modifying the transforms without knowing what you're doing is usually unwise, as you can make the image object inconsistent.

Author(s)

Jon Clayden <code@clayden.org>

References

The NIfTI-1 standard (http://www.nitrc.org/docman/view.php/26/64/nifti1.h) is the definitive reference on "xform" conventions.

Examples

```
im <- readNifti(system.file("extdata", "example.nii.gz", package="RNifti"))
xform(im)

# Remove the qform information
qform(im) <- structure(diag(4), code=0L)

# The same as above, since the sform is unmodified
xform(im)

# The identity matrix corresponds to RAS orientation
orientation(diag(4))
```

`$.niftiImage` *Access to metadata elements*

Description

These methods provide shorthand access to metadata elements from the NIfTI header corresponding to an image. The extraction version returns the corresponding element from the result of `niftiHeader`, while the replacement version calls `updateNifti` to replace it.

Usage

```
## S3 method for class 'niftiImage'
x$name

## S3 replacement method for class 'niftiImage'
x$name <- value
```

Arguments

x	A "niftiImage" object, internal or otherwise.
name	A string naming the field required.
value	A new value for the field.

Author(s)

Jon Clayden <code@clayden.org>

See Also

niftiHeader, updateNifti

Examples

```
im <- readNifti(system.file("extdata", "example.nii.gz", package="RNifti"))
print(im$descrip)
```

Index